4H-01

区分的微分可能関数と一般化自動微分

吉田 北斗 †　　久保田光一 ‡

中央大学大学院 理工学研究科 † 中央大学 理工学部 ‡

1 はじめに

非線形最適化問題を解く種々の反復解法は, その目的関数や制約式の微分情報に依存していることが多い. しかし, 工学問題ではしばしば区分的微分可能 (Piecewise Smooth) なモデルを用いることがあり, そういったモデルを表す目的関数は至るところで微分不可能となっているため, 従来の手法ではその微分不可能点周辺で軌道がジグザグになるなど収束が不規則になってしまう.

本研究では, Piecewise Smooth の中でも Lipschitz 連続なモデルに対する微分情報を用いた最適化手法の提案とその収束の安定性の評価を行う.

2 自動微分 [1]

2.1 基本演算

自動微分では, プログラム中で使用できる演算

- $+, -, \times, /$
- sqrt, pow, log, exp, sin, ...

を基本演算と呼ぶ. 以下では, この基本演算に, abs (絶対値関数), min/max (最小・最大値関数) も含める. また, これら基本演算の合成関数を基本演算の一つとすることもできる.

2.2 中間変数

自動微分では, 基本演算一つと一対一に対応する中間変数と呼ばれる変数に, その演算の実行結果を格納する.

$\boldsymbol{y} = f(\boldsymbol{x})$ $(f : \mathbb{R}^n \to \mathbb{R}^m)$ の自動微分による解釈は以下のようになる.

u_{i-n}	$=$	x_i	$i = 1, \ldots, n$
u_i	$=$	$\varphi_i(u_j)_{j \prec i}$	
y_i	$=$	$\varphi_i(u_j)_{j \prec i}$	$i = 1, \ldots, m$

$\varphi_\cdot$ は基本演算, $\boldsymbol{u}$ は中間変数となる. 中間変数には, その解釈順に順序関係がある.

2.3 微分係数の導出

順序に従って各中間変数の微分係数を計算していく, すなわち, 自動微分は合成関数の微分則, 鎖律 (chain rule) に従って微分係数を計算する.

$\frac{\partial u_{i-n}}{\partial x_j}$	$=$	$i == j \ ? \ 1 \ : \ 0$	$i = 1, \ldots, n, \ j = 1, \ldots, n$
$\frac{\partial u_i}{\partial x_j}$	$=$	$\Sigma_{k \prec i} \varphi_i'(u_k) \frac{\partial u_k}{\partial x_j}$	$j = 1, \ldots, n$
$\frac{\partial y_i}{\partial x_j}$	$=$	$\Sigma_{k \prec i} \varphi_i'(u_k) \frac{\partial u_k}{\partial x_j}$	$i = 1, \ldots, m, \ j = 1, \ldots, n$

2 階以上の高階微分係数についても自動微分で導出可能であるが, 本研究では利用しないため解説を省くことにする.

自動微分には Bottom-Up 型, Top-Down 型と呼ばれる算法があるが, 後述する理論はどちらの算法でも実現可能である.

3 Abs-Normal Form [2, 3]

3.1 Piecewise Smooth

プログラム中で Piecewise Smooth を表現する要素として abs, min/max, if 文などがあげられるが, これらには abs を用いた書き換えが知られている [4, 2]. 以下に min/max の例を示す.

$$\min(u, w) = (u + w - \mathrm{abs}(u - w))/2, \quad \max(u, w) = (u + w + \mathrm{abs}(u - w))/2$$

3.2 基本演算の区分線形化

各基本演算の増分の近似を以下のように考える.

$$\begin{aligned}
\Delta v_i &= \Delta v_j \pm \Delta v_k && \text{when} \quad v_i = v_j \pm v_k \\
\Delta v_i &= v_j \Delta v_k + v_k \Delta v_j && \text{when} \quad v_i = v_j \dot{v}_k \\
\Delta v_i &= \varphi_i'(v_j)_{j \prec i} \Delta(v_j)_{j \prec i} && \text{when} \quad v_i = \varphi_i(v_j)_{j \prec i} \neq |v_j| \\
\Delta v_i &= |v_j + \Delta v_j| - v_i && \text{when} \quad v_i = |v_j|
\end{aligned}$$

このとき, 関数 f の点 $\boldsymbol{x}$ での方向 $\Delta \boldsymbol{x}$ への区分線形近似を

$$f_{\mathrm{PL},\boldsymbol{x}}(\Delta \boldsymbol{x}) = f(\boldsymbol{x}) + \Delta f(\boldsymbol{x}; \Delta \boldsymbol{x}) \tag{1}$$

と表すことができる.

3.3 signature/switch

絶対値関数を含む方程式 $y = f(\boldsymbol{x})$ $(f : \mathbb{R}^n \to \mathbb{R})$ を, 自動微分によって以下のように解釈する.

w_{i-n}	$=$	x_i	$i = 1, \ldots, n$
z_i	$=$	$\varphi_i(w_j)_{j \prec i}$	
σ_i	$=$	$\mathrm{sign}(z_i)$	$i = 1, \ldots, s$
w_i	$=$	$\sigma_i z_i = \lvert z_i \rvert$	
y	$=$	$\varphi_s(w_j)_{j \prec s}$	

$\varphi_\cdot$ は $|\cdot|$ 以外の基本演算, s は f の解釈中に評価した $|\cdot|$ の個数, $\boldsymbol{z}, \boldsymbol{w}$ が中間変数となる.

$$\begin{aligned}
&\text{switching vector} \quad \boldsymbol{z} \equiv \boldsymbol{z}(\boldsymbol{x}) \in \mathbb{R}^s \\
&\text{signature vector} \quad \boldsymbol{\sigma} \equiv \boldsymbol{\sigma}(\boldsymbol{x}) \equiv \mathrm{sign}(\boldsymbol{z}(\boldsymbol{x})) \in \{-1, 0, 1\}^s
\end{aligned}$$

に着目し, $\Sigma \equiv \mathrm{diag}(\boldsymbol{\sigma})$ とおくと

$$\boldsymbol{z} \equiv \boldsymbol{z}(\boldsymbol{w}) \equiv \Phi(\boldsymbol{x}, \boldsymbol{w}) \quad (\Phi : \mathbb{R}^{n+s} \to \mathbb{R}^s), \tag{2}$$

$$y \equiv y(\boldsymbol{w}) \equiv \tilde{f}(\boldsymbol{x}, \boldsymbol{w}) \quad (\tilde{f} : \mathbb{R}^{n+s} \to \mathbb{R}) \tag{3}$$

と定義される. このような方程式の再解釈を Abs-Normal Form と呼ぶ.

3.4 方程式の区分線形化

式 (2), (3) より

$$\Delta \boldsymbol{z} = \frac{\partial \Phi}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} + \frac{\partial \Phi}{\partial \boldsymbol{w}} \Delta \boldsymbol{w},$$

$$\Delta y = \frac{\partial \tilde{f}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} + \frac{\partial \tilde{f}}{\partial \boldsymbol{w}} \Delta \boldsymbol{w}$$

を得る. これを行列の形でまとめると以下のようになる.

$$\begin{bmatrix} \Delta \boldsymbol{z} \\ \Delta y \end{bmatrix} = \begin{bmatrix} \frac{\partial \Phi}{\partial \boldsymbol{x}} & \frac{\partial \Phi}{\partial \boldsymbol{w}} \\ \frac{\partial \tilde{f}}{\partial \boldsymbol{x}} & \frac{\partial \tilde{f}}{\partial \boldsymbol{w}} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} Z & L \\ \boldsymbol{a}^T & \boldsymbol{b}^T \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{w} \end{bmatrix}$$

$$Z \in \mathbb{R}^{s \times n}, \ L \in \mathbb{R}^{s \times s}, \ \boldsymbol{a} \in \mathbb{R}^n, \ \boldsymbol{b} \in \mathbb{R}^s$$

Piecewise Smooth Function and Generalized Algorithmic Differentiaion

† Hokuto YOSHIDA, Graduate School of Science and Engineering, Chuo University

‡ Koichi KUBOTA, Faculty of Science and Engineering, Chuo University

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

中間変数が半順序関係を持つことにより, L は対角成分がゼロの下三角行列となる. 区分線形化を踏まえてこれを整理すると

$$\Delta \boldsymbol{z} = (I - L\Sigma)^{-1} Z \Delta \boldsymbol{x},$$

$$\Delta y = (\boldsymbol{a}^T + \boldsymbol{b}^T \Sigma (I - L\Sigma)^{-1} Z) \Delta \boldsymbol{x}$$

を得る. $\Delta f(\boldsymbol{x}; \Delta \boldsymbol{x}) \equiv \Delta y$ とすれば式 (1) より, Piecewise Smooth な f に対しても $f_{\mathrm{PL},\boldsymbol{x}}$ が計算可能となる.

4 最急降下法の拡張 [5, 6]

4.1 Selection Function

ある signature vector $\boldsymbol{\sigma}$ を定めたときの $f_{\mathrm{PL},\boldsymbol{x}}$ を $f_{\boldsymbol{\sigma},\boldsymbol{x}}$ と表し, これを Selection Function と呼ぶ. 以下に Selection Function の例を示す.

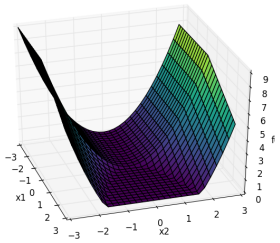

図 1 $f(x_1, x_2) = (x_2^2 - (x_1)_+)_+$

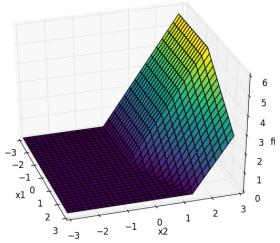

図 2 $f_{\mathrm{PL},(1,1)}$

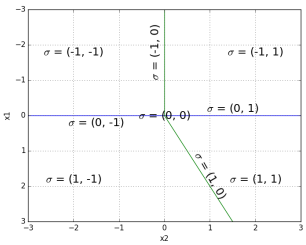

図 3 $\boldsymbol{\sigma}$ による領域分割

提案手法ではこの Selection Function 上で $\boldsymbol{\sigma}$ の切り替わりに注目しつつ, もとの関数 f での降下方向を近似しながら近似解を探索する.

4.2 Taylor 展開による近似

目的関数 f 上で $\boldsymbol{x}$ から方向 $\Delta \boldsymbol{x}$ だけ遷移したときの関数値の近似を Taylor 展開の 2 次の項までの展開で与える.

$$f(\boldsymbol{x} + \Delta \boldsymbol{x}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \Delta \boldsymbol{x} + \frac{1}{2} \Delta \boldsymbol{x}^T \nabla^2 f(\boldsymbol{x}) \Delta \boldsymbol{x}$$

いま, f は Piecewise Smooth であり, $\nabla f, \nabla^2 f$ が求まらない場合がある. ここで, Abs-Normal Form による線形近似により

$$f(\boldsymbol{x} + \Delta \boldsymbol{x}) = f_{\mathrm{PL},\boldsymbol{x}}(\Delta \boldsymbol{x}) + \frac{1}{2} \Delta \boldsymbol{x}^T \nabla^2 f(\boldsymbol{x}) \Delta \boldsymbol{x} \tag{4}$$

のように, Taylor 展開を更に近似した式を得る. しかし, これに加えて $\nabla^2 f$ の良い近似を考える必要がある.

また, この公式を用いた連続最適化で得られた近似解は, 基本的には $f_{\mathrm{PL},\boldsymbol{x}}$ 上での局所最適であるが, もとの目的関数 f が区分線形の場合には f 上で局所最適であることが, 区分線形かつ凸である場合には f 上で大域最適であることが知られている [5].

4.3 提案手法

アルゴリズム 1 の PSMIN により最小化を行う. 手続き中で呼び出されている PLMIN, SHORTEST はそれぞれ, $f_{\mathrm{PL},\boldsymbol{x}}$ 上での最小化と f 上での降下方向の計算, 接バンドル G との最短距離方向の計算を行う手続きである.

式 (4) の第 2 項を近似する方法として, 準ニュートン法で知られる BFGS 公式を採用する.

Algorithm 1 Piecewise Smooth Minimization

procedure PSMIN($\boldsymbol{x}$)
  $B_0 \leftarrow I$
  for $k \leftarrow 1, 2, \cdots$ **do**
    $\Delta \boldsymbol{x}_k \leftarrow \mathrm{PLMIN}(B_k, \boldsymbol{x}_k)$
    if $||\Delta \boldsymbol{x}_k|| = 0$ **then** STOP;
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \Delta \boldsymbol{x}_k$
    $B_{k+1} \leftarrow B_{k+1}^{\mathrm{BFGS}}$
  end for
  return $\boldsymbol{x}_k, f(\boldsymbol{x}_k)$
end procedure
procedure PLMIN($B, \boldsymbol{x}$)
  $\Delta \boldsymbol{x} \leftarrow \boldsymbol{0}$
  $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}(\boldsymbol{x})$
  $G \leftarrow \emptyset$
  loop
    $\delta \boldsymbol{x} \leftarrow \underset{\delta \boldsymbol{x}}{\mathrm{argmin}} \left\{ f_{\sigma,\boldsymbol{x}}(\delta \boldsymbol{x}) + \frac{1}{2} \delta \boldsymbol{x}^T B^{-1} \delta \boldsymbol{x} \middle| \Sigma \boldsymbol{z} \geq 0 \right\}$
    $\Delta \boldsymbol{x} \leftarrow \Delta \boldsymbol{x} + \delta \boldsymbol{x}$
    $\boldsymbol{g} \leftarrow \nabla f_{\sigma,\boldsymbol{x}}$
    $G \leftarrow G \cup \{\boldsymbol{g}\}$
    $\boldsymbol{d} \leftarrow -\mathrm{SHORTEST}(B^{-1} \delta \boldsymbol{x}, G)$
    if $||\boldsymbol{d}|| = 0$ **then** STOP;
    if $(\boldsymbol{g} + B_k^{-1} \Delta \boldsymbol{x})^T \boldsymbol{d} > 0$ **then**
      repeat
        $\boldsymbol{d} \leftarrow \mathrm{SHORTEST}(B^{-1} \delta \boldsymbol{x}, G)$
        $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}(\boldsymbol{x}_k + \boldsymbol{d})$
        $\boldsymbol{g} \leftarrow \nabla f_{\sigma,\boldsymbol{x}}$
        $G \leftarrow G \cup \{\boldsymbol{g}\}$
      until $(\boldsymbol{g} + B^{-1} \delta \boldsymbol{x})^T \boldsymbol{d} \leq -||\boldsymbol{d}||^2$
      $G \leftarrow \emptyset$
      $B \leftarrow I$
    end if
    $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}(\boldsymbol{x} + \boldsymbol{d})$
  end loop
  return $\Delta \boldsymbol{x}$
end procedure
procedure SHORTEST($\boldsymbol{x}, G$)
  return $\underset{\boldsymbol{d}}{\mathrm{argmin}} \left\{ ||\boldsymbol{d}|| \middle|_{\boldsymbol{g}_j \in G,\ \beta_j \geq 0,\ \Sigma_{j=1}^m \beta_j = 1}^{\boldsymbol{d} = \Sigma_{j=1}^m \beta_j \boldsymbol{g}_j - B_k^{-1} \delta \boldsymbol{x},} \right\}$
end procedure

5 数値実験

検証実験については当日の資料参照.

6 今後の課題

区分的微分可能関数に対する 2 階微分のより良い近似を, 自動微分の算法, その特徴の中から考察すること, また, 制約付き問題や方程式系への応用を今後の課題とする.

参考文献

[1] 久保田 光一, 伊理 正夫, アルゴリズムの自動微分と応用. Automatic differentiation of algorithms and applications, 東京, Japan: コロナ社, 1998.7 1998.
[2] A. Griewank, "On stable piecewise linearization and generalized algorithmic differentiation," *Optimization Methods Software*, vol. 28, pp. 1139–1178, Dec. 2013.
[3] A. Griewank, A. Walther, S. Fiege, and T. Bosse, "On Lipschitz optimization based on gray-box piecewise linearization," *Mathematical Programming*, vol. 158, no. 1, pp. 383–415, 2016.
[4] S. Scholtes, *Piecewise Differentiable Functions*, pp. 91–111. New York, NY: Springer New York, 2012.
[5] A. Griewank, J.-U. Bernt, M. Radons, and T. Streubel, "Solving piecewise linear equations in abs-normal form," *Documenta Mathematica*, vol. Optimization Stories, 2015.
[6] F. Clarke, *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990.

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.